AMENDMENTS TO THE CLAIMS:

This listing of claims replaces all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A configurable cyclic redundancy check (CRC) ~~CRC~~ calculation engine comprising:

a CRC unit;

at least one polynomial storage device to provide a polynomial to the CRC unit; ~~and~~

at least one residue storage device to provide a residue ~~polynomial~~ to the CRC unit, wherein the CRC unit is adapted to determine a CRC value for received data using said polynomial and the residue; and

an input random access memory (RAM) coupled to the CRC unit and the at least one polynomial storage, the RAM input being configured to provide the polynomial to the at least one polynomial storage.

2. (Original) The configurable CRC calculation engine of claim 1 wherein said CRC unit is adapted to write said CRC value into said residue storage device.

3. (Cancelled)

4. (Original) The configurable CRC calculation engine of claim 1 wherein said configurable CRC calculation engine includes a plurality of processing contexts, each of said plurality of processing contexts corresponding to one of said at least one polynomial storage device and to one of said at least one residue storage device.

5. (Original) The configurable CRC calculation engine of claim 1 wherein said at least one polynomial storage device is loaded at initialization time.

6. (Original) The configurable CRC calculation engine of claim 1 wherein said at least one polynomial storage device is loaded for each new packet of data.

7. (Currently Amended) A method of determining ~~calculating~~ a cyclic redundancy check (CRC) ~~CRC~~ value comprising:

receiving a polynomial associated with a packet of data;

receiving a residue associated with ~~a~~ the packet of data provided by a CRC calculation engine;

receiving a block of data, said block of data comprising a portion of said packet;

~~calculating~~ determining a CRC value, using the CRC calculation engine, for the block of data using said polynomial and said residue; and

storing the CRC value.

8. (Original) The method of claim 7 wherein said block of data comprises a part of a packet of data and wherein said method further comprises determining whether there are remaining blocks of data for the packet and when there are remaining blocks of data for the packet then loading the next block of data for the packet and calculating a CRC value for said next block of data.

9. (Currently Amended) The method of claim 8 wherein said determining ~~calculating~~ a CRC value for said next block of data includes using a residue from a CRC calculation for a prior block of data.

10. (Original) The method of claim 7 wherein said receiving a polynomial further comprises initializing a residue to zero.

11. (Original) The method of claim 7 wherein an initial non-zero value is loaded into the residue.

12. (Original) The method of claim 7 wherein said loading a polynomial into a context is done at initialization time.

13. (Original) The method of claim 7 wherein said loading a polynomial into a context is done for a new packet of data.

14. (Currently Amended) A cyclic redundancy check (CRC) ~~CRC~~ calculation engine comprising:

an input data storage unit having a plurality of outputs;

a polynomial storage device having an output;

a plurality of single data bit processors coupled together serially and coupled to a respective one of said input data storage unit plurality of outputs and to said polynomial storage device output; and

a remainder storage element coupled to a first one of the plurality of single data bit processors and configured to provide a residue to the first one of the single bit data processors ~~one of said plurality of single data bit processors~~.

15. (Currently Amended) The CRC calculation engine of claim 14 wherein ~~a first of said plurality of single bit data processors~~ the first one of the single data bit processors is configured to operate on a least significant bit ~~operates on a Least Significant Bit~~ (LSB) of data stored in the ~~said~~ input data storage unit.

16. (Original) The CRC calculation engine of claim 14 wherein each of said single data bit processors performs an exclusive or function.

17. (Original) A programmable CRC calculation engine comprising:

a first stage adapted to receive data and first stage configuration bits, and to determine an interim CRC value; and

a second stage coupled to said first stage, said second stage adapted to receive said interim CRC value, second stage configuration bits and an end of data signal and to determine a CRC value for said data.

18. (Original)The programmable CRC engine of claim 17 wherein said first stage comprises:

a plurality of AND gates:

at least one XOR tree having at least one input coupled to an output of at least one AND gate; and

a register having at least one input coupled to at least one output of said at least one XOR tree.

19. (Original) The programmable CRC engine of claim 17 wherein said second stage comprises:

a plurality of AND gates;

at least one XOR tree having at least one input coupled to an output of at least one AND gate; and

a register having at least one input coupled to at least one output of said at least one XOR tree.

20. (Original) The programmable CRC engine of claim 19 wherein said second stage further comprises at least one multiplexor coupled to an input of at least one AND gate, said at least one multiplexor having an input coupled to said end of data signal.

21. (Original) A method of calculating a CRC value comprising:

determining a first XOR tree;

loading data, a remainder and first stage configuration bits into said first XOR tree;

calculating an interim CRC value with said first XOR tree;

determining a second XOR tree;

loading said interim CRC value, an end of data value and second stage configuration bits into said second XOR tree;

calculating a CRC value for said data with said second XOR tree.

22. (Original) The method of claim 21 further comprising determining said first stage configuration bits.

23. (Original) The method of claim 21 further comprising determining said second stage configuration bits.

24. (Original) The method of claim 21 wherein said determining a first XOR tree comprises deriving said first XOR tree from said remainder, said first stage configuration bits and said data.

25. (Original) The method of claim 24 wherein said deriving said first XOR tree includes providing an XOR function for each bit of said remainder and said data and for the first stage configuration bits and said data.

26. (Original) The method of claim 22 wherein determining said first stage configuration bits comprises setting corresponding first stage configuration bits to zero that don't appear in a polynomial used with said data to determine said CRC.

27. (Original) The method of claim 21 wherein said determining said second XOR tree comprises deriving an XOR tree from said interim CRC value and said second stage configuration bits.

28. (Original) The method of claim 27 said deriving said second XOR tree includes providing an XOR function for each bit of said remainder and a predetermined set of said second stage configuration bits.

29. (Original) The method of claim 28 wherein said predetermined set of second stage configuration bits is determined from said end of data value.

30. (New) The CRC calculation engine of claim 15 wherein the remainder storage element is coupled to a second one of the plurality of single data bit processors and configured to receive the residue from the second one of the single data bit processors.